

ConceptBase - Eine Metadatenbank für konzeptuelle Modellierungsaufgaben

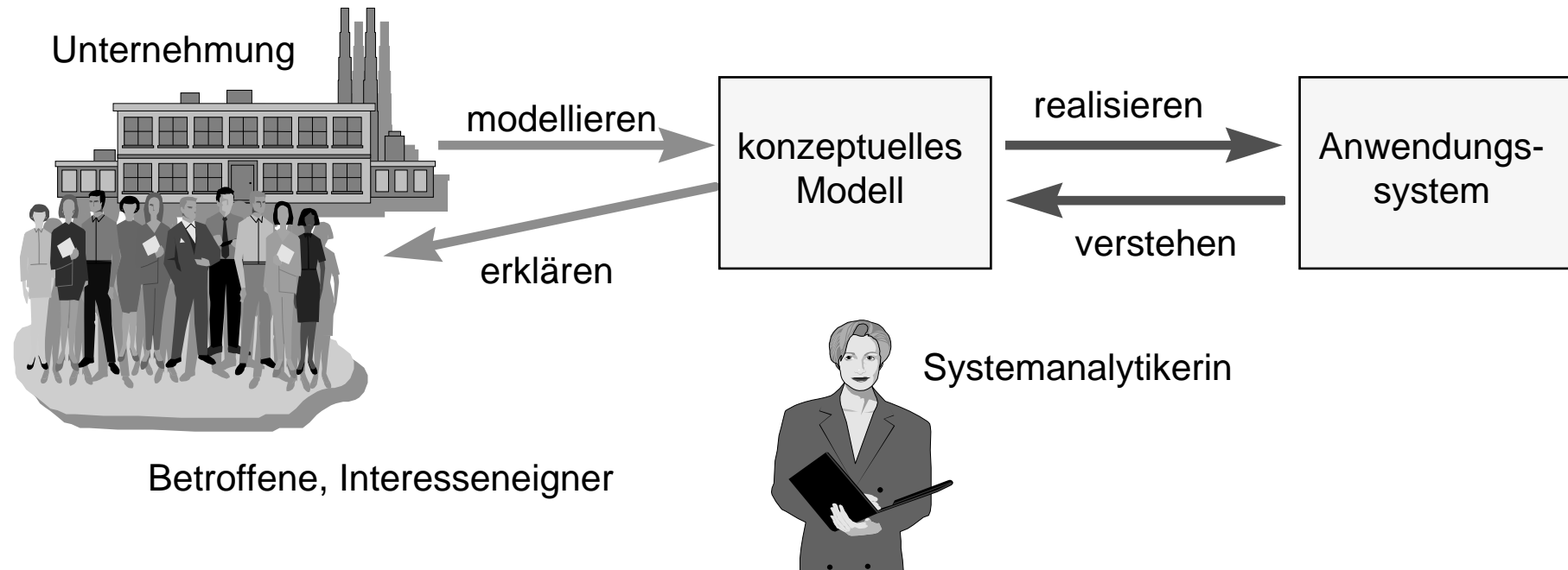
Manfred Jeusfeld

RWTH Aachen, Informatik 5, Ahornstr. 55, 52056 Aachen
jeusfeld@informatik.rwth-aachen.de
<http://www-i5.informatik.rwth-aachen.de/mjf>



Lehrstuhl Informatik V
(Informationssysteme)
Prof. Dr. M. Jarke

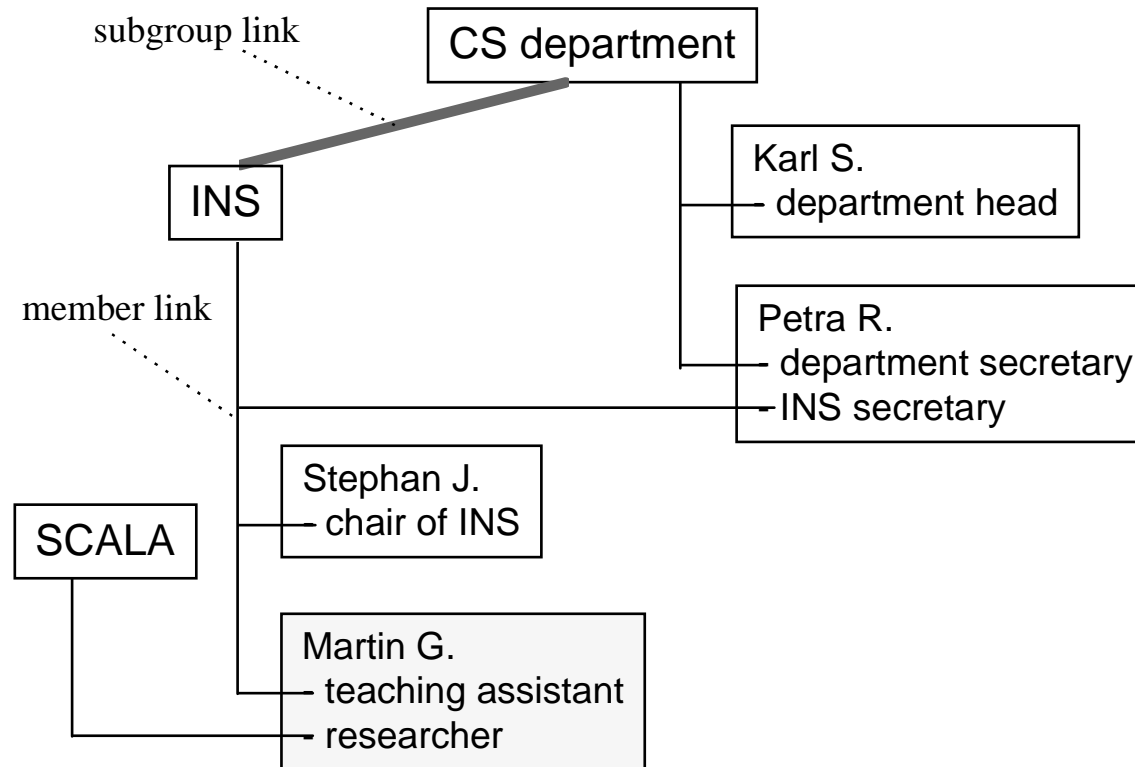
Der doppelte Zweck der konzeptuellen Modellierung



Konzeptuelle Modell ist

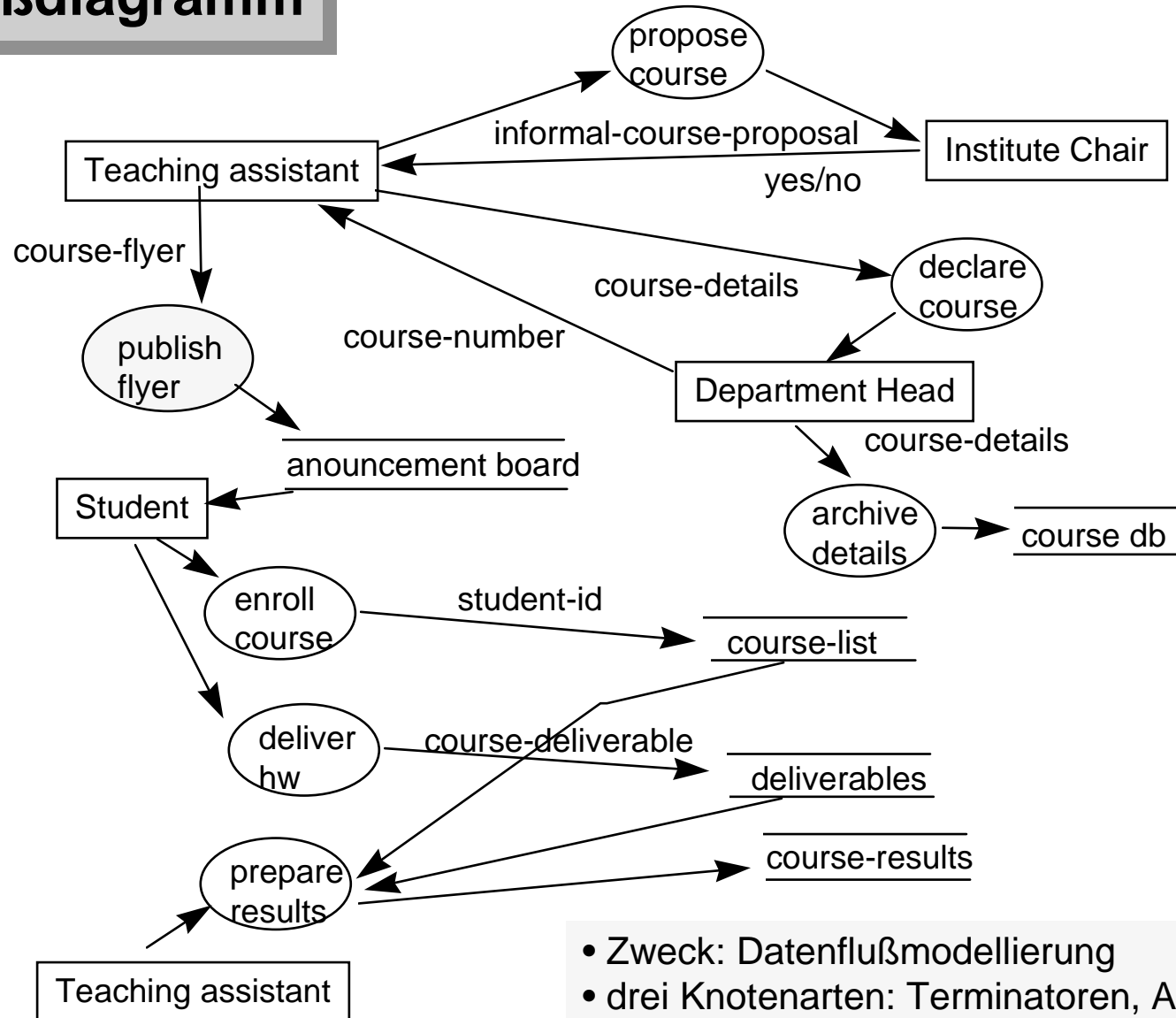
- a) ein abstraktes Modell eines Sachverhalts (linke Seite)
- b) ein abstraktes Modell eines Anwendungssystems (rechte Seite)

Organigramm



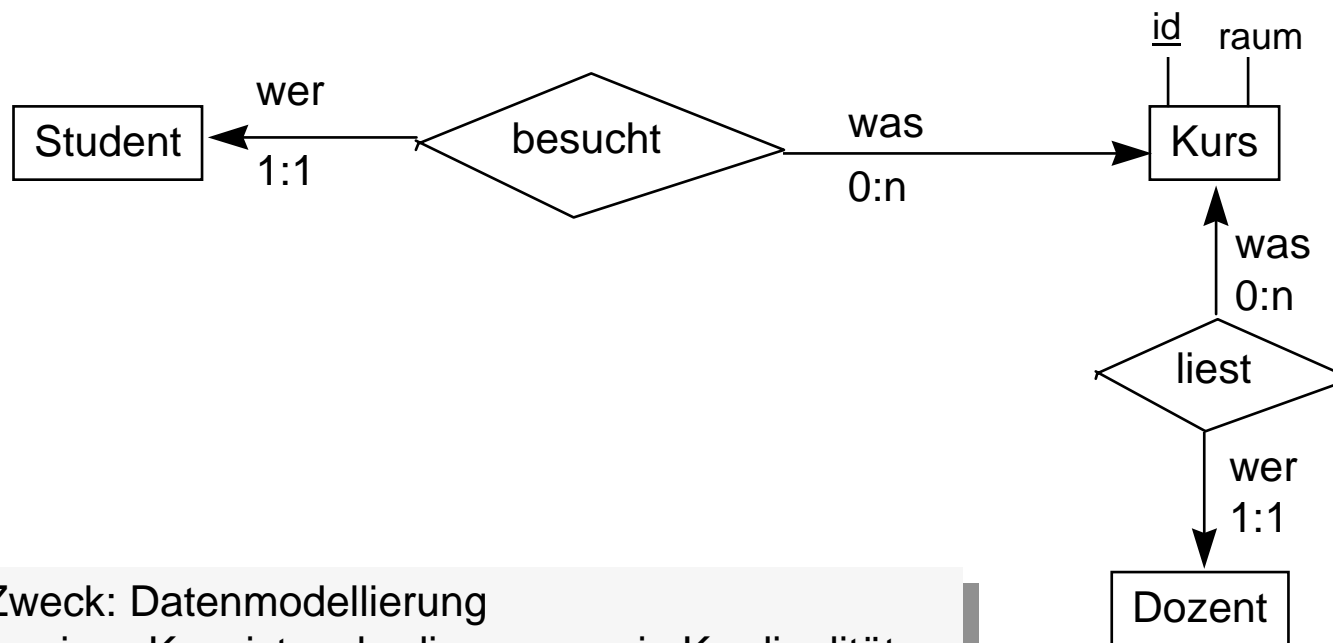
- Zweck: Modellierung der Organisationsstruktur
- Gruppen, Mitgliedschaft in Gruppen

Datenflußdiagramm



- Zweck: Datenflußmodellierung
- drei Knotenarten: Terminatoren, Aktivitäten, Speicher
- Interpretation gewollt unscharf
- Bezug zu Data Dictionary und ERD

Entity-Relationship-Diagramm



- Zweck: Datenmodellierung
- gewisse Konsistenzbedingungen via Kardinalitäten und Schlüsselattributen
- Dutzende Varianten (mit/ohne ISA, PARTOF)

Weitere konzeptuelle Modellierungssprachen

- EPK: Ereignis-Prozeß-Kette
- OMT: Object Modeling Technique
- Flußdiagramme
- Petrinetze
- Workflow-Diagramme
- IDEF1X-Diagramme (SADT)
- ...

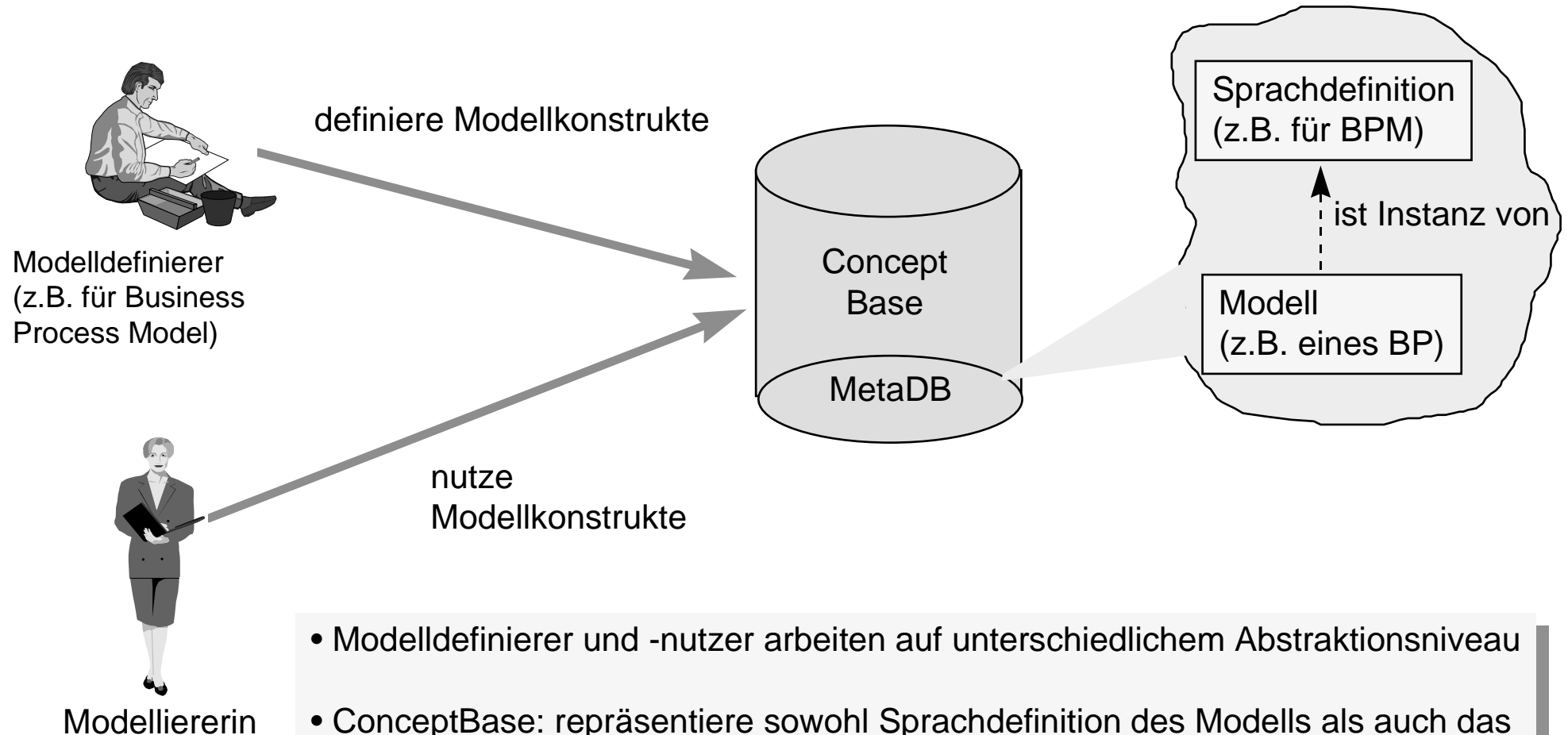
Es gibt eine Vielzahl solcher Modellierungssprachen.
Gemeinsames Kennzeichen: graphische Notation

Oft wird eine Kombination von 2-5 Modellierungssprachen in der Systemanalyse eingesetzt. Beachte: Die Diagramme stehen untereinander in Beziehung.

Probleme:

- 1) Festlegung der Semantik der Diagramme.
- 2) Management des Prozesses der konzeptuellen Modellierung; Schwachstellenanalyse

Die ConceptBase-Idee: Metamodellierung



- Modelldefinierer und -nutzer arbeiten auf unterschiedlichem Abstraktionsniveau
- ConceptBase: repräsentiere sowohl Sprachdefinition des Modells als auch das konzeptuelle Modell als Objekte der 'Metadatenbank'
- explizite Repräsentation der Abstraktionsbeziehung 'ist Instanz von'

Die “Lösung”: O-Telos mit ConceptBase

Was kennzeichnet eine konzeptuelle Modellierungssprache?

- graphische Notation: repräsentiere Knoten und Kanten
- Axiome über zulässige Modelle: repräsentiere logische Formeln
- Axiome über die Eigenschaften von Instanzen der Modelle: ‘Metaformeln’

Was kennzeichnet den Modellierungsprozeß?

- mehrere Modellierer: ConceptBase muß mehrbenutzerfähig sein
- unterschiedliche Sichtweisen der Benutzer: erlaube ‘Partialmodelle’
- Fehler, Unvollständigkeiten: biete komfortable Analysefähigkeit



Nächste Folien:

- O-Telos als Metamodellierungssprache
- ConceptBase: ein Metadatenbanksystem auf Basis von O-Telos
- Erfahrungsbericht: Business Process Modelling

O-Telos: das kleinste objektorientierte Datenmodell der Welt

Def:

Sei $OB = \{P(o, x, n, y) \mid o, x, y \text{ in ID, } n \text{ in LABEL}\}$ eine endliche Menge sogenannter Objekte. Ferner sei IC eine Menge geschlossener prädikatenlogischer Formeln über dem 'Basisprädikat' $P(o, x, n, y)$ und R eine Menge von Hornklausenformeln über dem Basisprädikat. Dann heißt (OB, IC, R) eine Objektbank (in O-Telos). Die Menge IC heißt auch Menge der Integritätsbedingungen und die Menge R auch Menge der deduktiven Regeln.

Die Semantik ist als Fixpunktsemantik definiert, wobei die Integritätsbedingungen als Kurzschreibweisen für deduktive Regeln mit Folgerungsprädikat 'inconsistent' aufgefaßt werden (siehe Ceri/Gottlob/Tanca: Logic and databases, Springer-Verlag).

Konsequenz: eindeutige Semantik, effiziente Auswertung, Nähe zu deduktiven Datenbanken

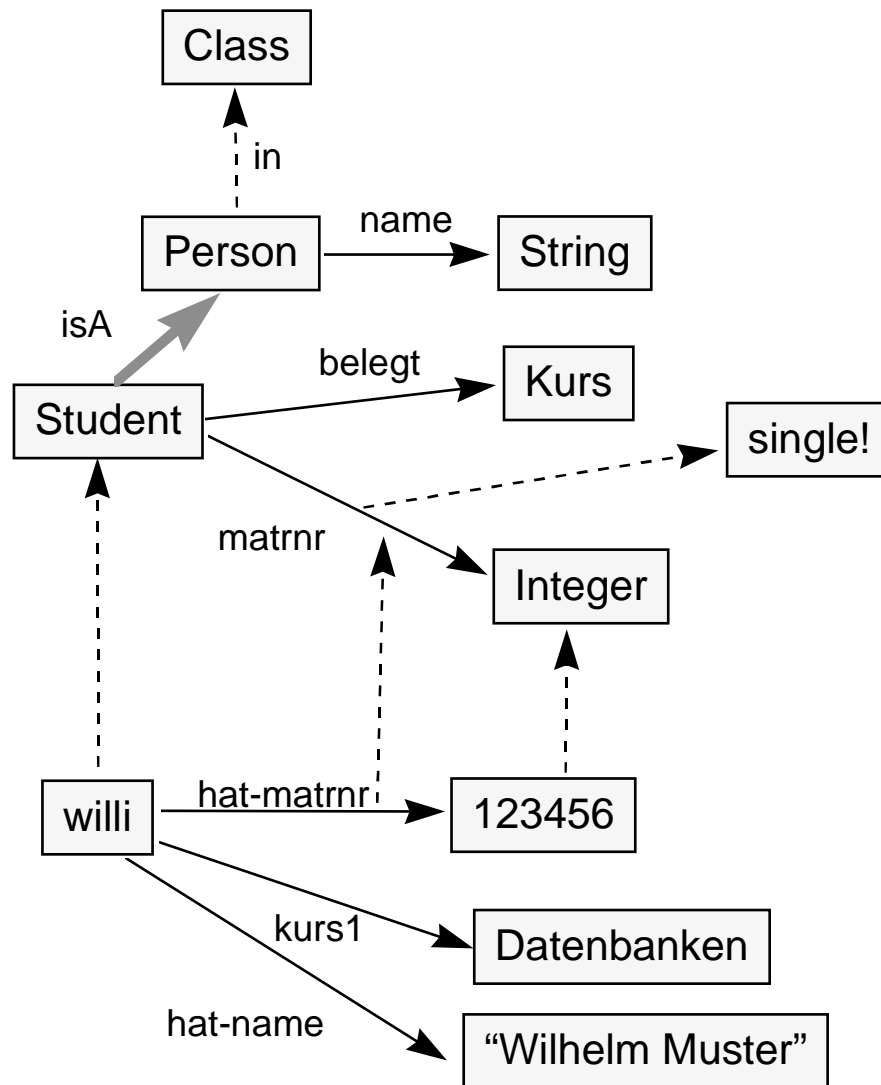
Frage: Wieso nennt man so etwas 'Objektbank'???

Antwort: nächste Folien

Frage: Was ist mit den vorhin erwähnten Axiomen?

Antwort: Die werden als deduktive Regeln und Integritätsbedingungen repräsentiert!

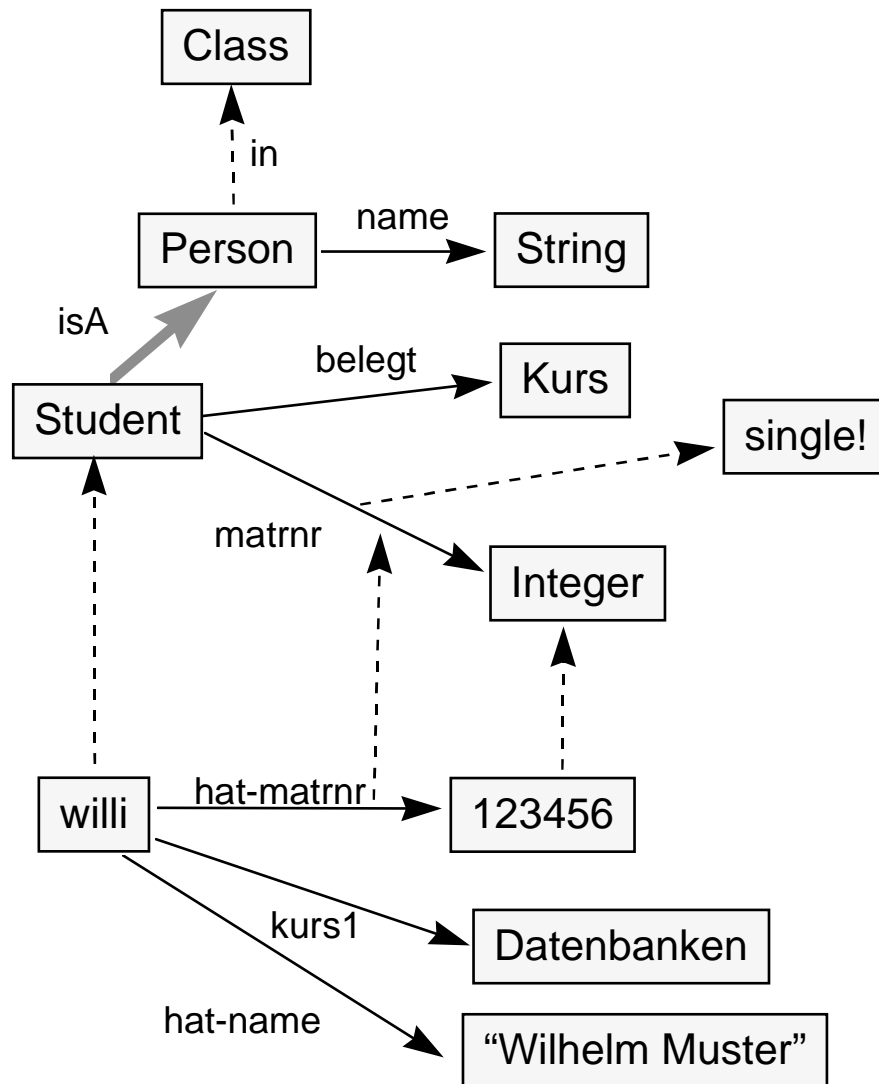
O-Telos: graphische Sicht und ihre Interpretation



```

P(#Cl,#Cl,Class,#Cl)
P(#P,#P,Person),#P)
P(#St,#St,Student,#St)
P(#w,#w,willi,#w)
P(#si,#si,single!,#si)
...
P(#at1,#P,name,#Str)
P(#at2,#St,matrnr,#Int)
P(#at3,#w,hat-matrnr,#123456)
...
P(#in1,#P,in,#Cl)
P(#in2,#at2,in,#si)
P(#in3,#w,in,#St)
P(#in4,#at3,in,#at2)
...
P(#isa1,#St,isA,#P)
    
```

O-Telos: Integritätsbedingungen und Regeln



forall o, x, c $P(o, x, \text{in}, c) \implies (x \text{ in } c)$

forall o, c, d $P(o, c, \text{isA}, d) \implies (c \text{ isA } d)$

forall c, d, e $(c \text{ isA } d) \text{ and } (d \text{ isA } e) \implies (c \text{ isA } e)$

forall x, c, d $(x \text{ in } c) \text{ and } (c \text{ isA } d) \implies (x \text{ in } d)$

forall o, x, n, y, p, c, m, d
 $P(o, x, n, y) \text{ and } (o \text{ in } p) \text{ and } P(p, c, m, d) \implies (x \text{ m/n } y)$

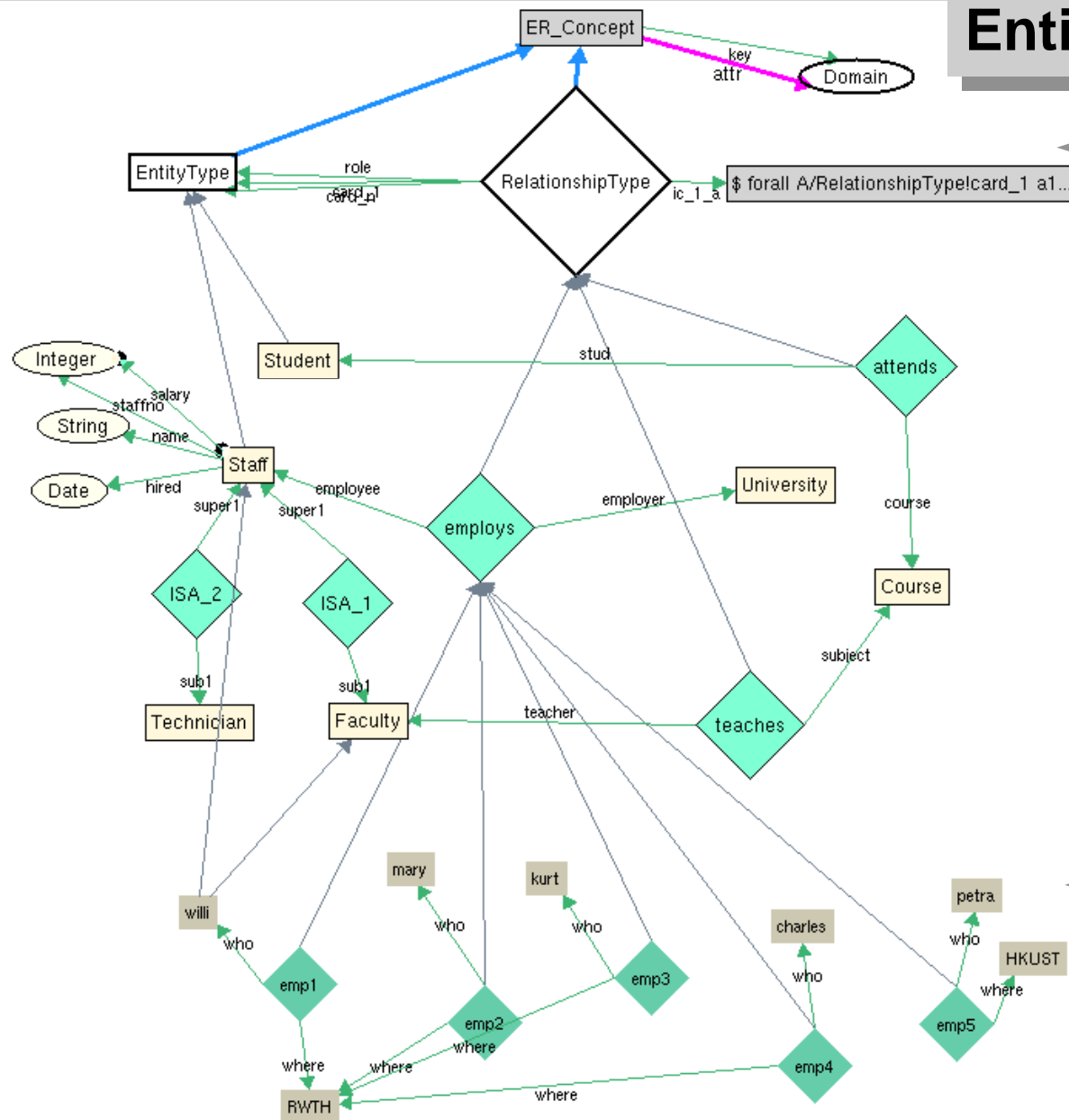
forall x, m, n, y $(x \text{ m/n } y) \implies (x \text{ m } y)$

forall o, x, n, y, p $P(o, x, n, y) \text{ and } (o \text{ in } p) \implies \text{exists } c, m, d \text{ } P(p, c, m, d) \text{ and } (x \text{ in } c) \text{ and } (y \text{ in } d)$

forall p, c, m, d, x $(p \text{ in } \#si) \text{ and } (x \text{ in } x) \text{ and } P(p, c, m, d) \implies (\text{forall } y, z \text{ } (y \text{ in } d) \text{ and } (z \text{ in } d) \text{ and } (x \text{ m } y) \text{ and } (x \text{ m } z) \implies (y = z))$

Solche Formeln sind in ConceptBase benutzerdefinierbar!

Beispiel: Darstellung des Entity-Relationship-Modells



Metamodell des ER-Modells in O-Telos

Ein konzeptuelles ER-Schema als Instanz des Metamodells

Beispielinstanzen des obigen ER-Schemas

in

in

Axiome vom ER-Modell in O-Telos

strukturelle
Eigenschaften

Axiom über die
Semantik von card_1

Browsen im Metamodell

Beachte: diese Definitionen sind
jederzeit anpaßbar ("alles ist
ein Objekt in ConceptBase")!!!

The screenshot shows the CBworkbench application window. The main text area contains the following O-Telos axiom for `RelationshipType`:

```
Individual RelationshipType in Class,Concept isA ER_Concept with
  attribute
    role : EntityType;
    card_1 : EntityType;
    card_n : EntityType
  graphtype,attribute
    gt : MetaRelationshipType_GT
  constraint,attribute
    ic_1_a : $ forall A/RelationshipType!card_1 a1,r/Proposition
      (a1 in A) and From(a1,r)
      ==> (forall a2/VAR (a2 in A) and From(a2,r) ==> (a1 == a2
    ) ) $
end
```

Below the main text area is a 'Protocol:' section with the following log messages:

```
Load model:
/home/jeusfeld/ERD/ERD-Instances.sml
Result: Ok
Model succesfully loaded !

Load object:
RelationshipType
Result: Answer
Individual RelationshipType in C
  attribute
    role : EntityType;
    card_1 : EntityType;
    card_n : EntityType
  graphtype,attribute
    gt : MetaRelationshipType_G
  constraint,attribute
    ic_1_a : $ forall A/Relatio
      (a1 in A)
      ==> (forall a2/v
    2) ) $
end
```

Overlaid on the bottom right is a 'Display Instances:' window. It shows a list of classes under the 'Class:' header:

- Domain
- ER_Concept
- EntityType
- IsaType
- RelationshipType

Below this list is a 'History:' section showing 'Concept'.

At the bottom of the CBworkbench window, there is a status bar with 'connected', 'rollback: Now', and '4 of 4'.

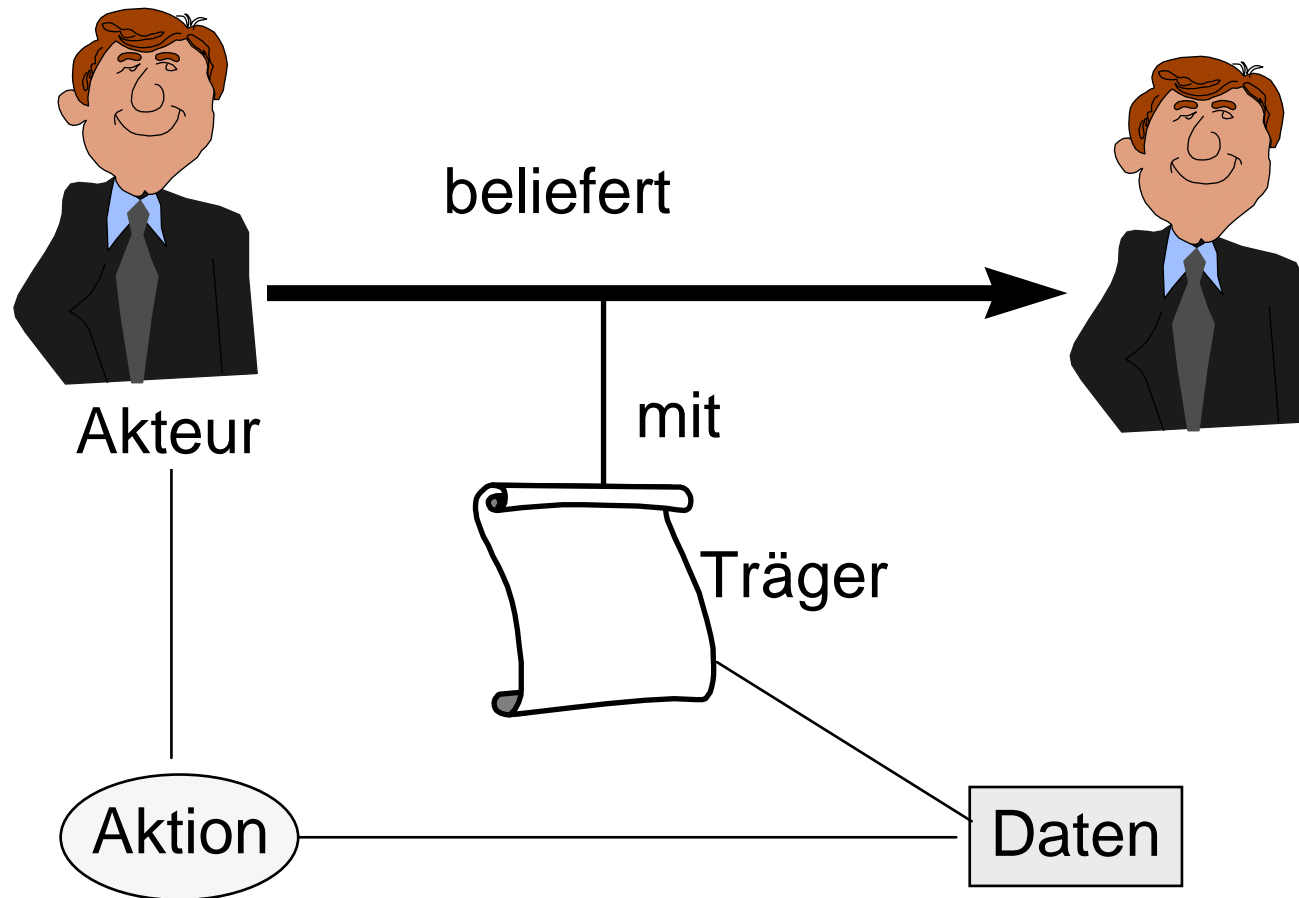
Anwendungsbeispiel: Analyse von Geschäftsprozessen

- konkretes Kooperationsprojekt mit einer mittelständischen Unternehmensberatung (USU AG)
- USU äußerte Mißfallen an Standardwerkzeugen wie ERD, ARIS, da zu starre Konzepte und inkompatibel mit der Vorgehensweise bei Kundenprojekten von USU
- ideale Validierungsstudie für ConceptBase, da hier sowohl die Sprachdefinition als auch deren Anwendung von Anfang an durchgezogen werden konnte

Lösungsidee

- Bilde ein Modell für die Informationsflüsse der Unternehmung
 - ➔ Was ist die geeignete Modellierungssprache?
SADT? ARIS?
- Analysiere dieses Modell, um Schwachstellen zu entdecken
 - ➔ Mit welchem Werkzeug findet man solche Schwachstellen?
Manuell? Datenbank?

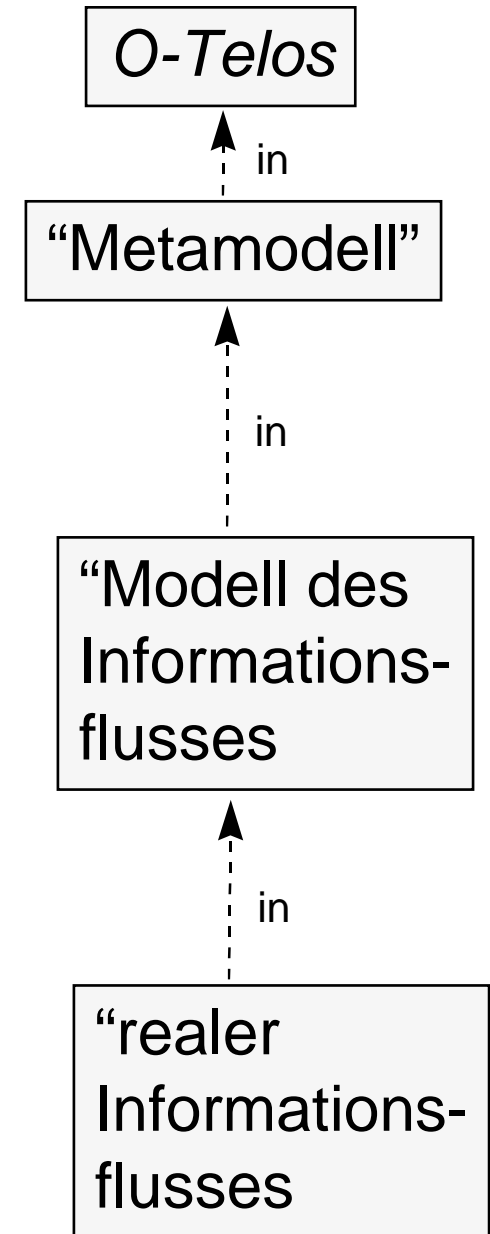
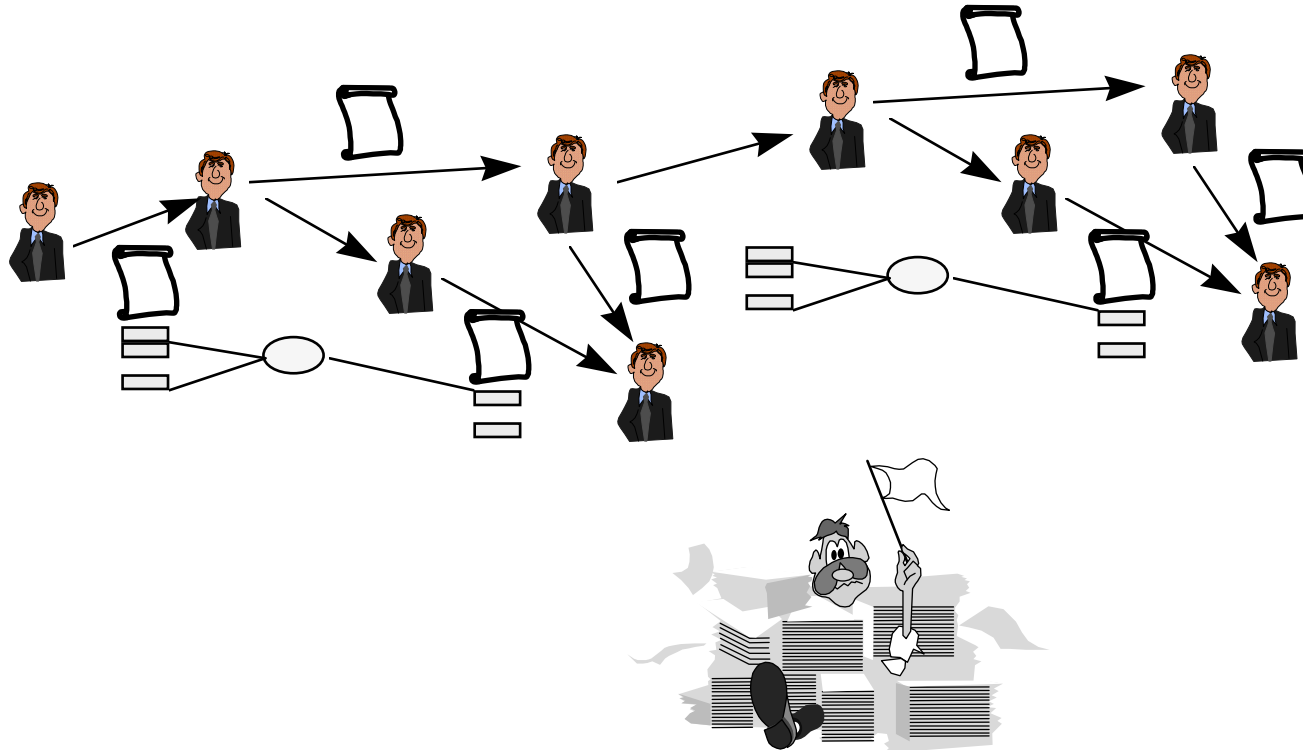
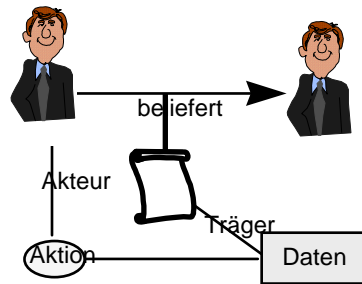
Metamodell für Informationsflüsse



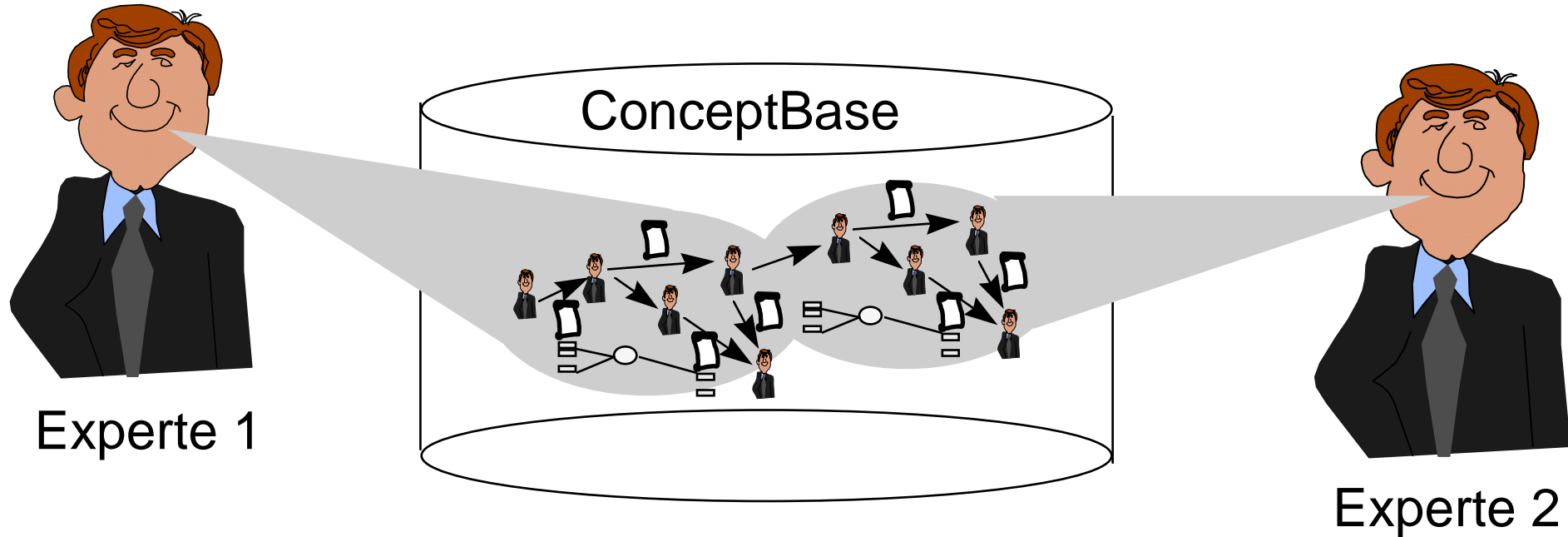
(Nissen, Jeusfeld, Jarke, Zemanek, Huber 96)

Beachte: Dieses Metamodell wurde in Kooperation mit USU innerhalb von 7 Tagen vollständig definiert

Definition der Modellierungssprache

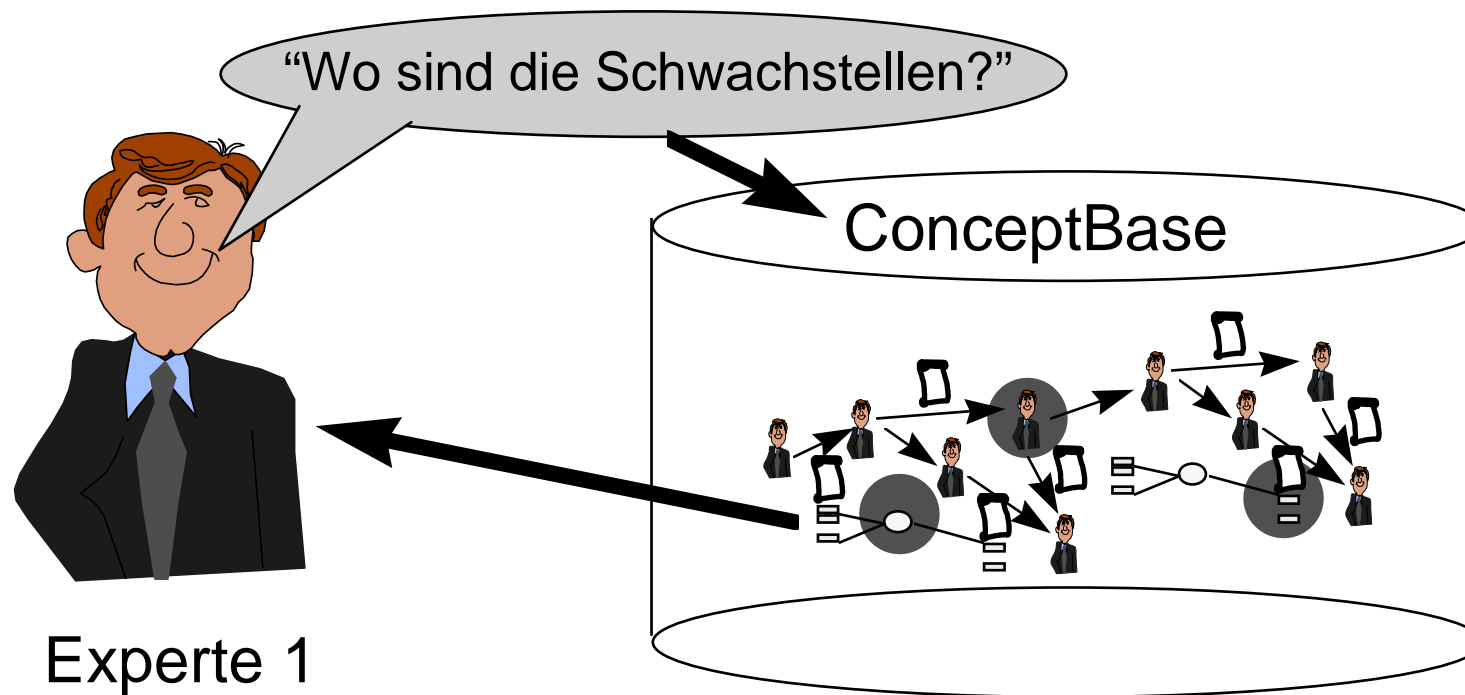


Modellierung mit ConceptBase



- Experten definieren die Informationsflüsse (TELL)

Modellvalidierung mit ConceptBase



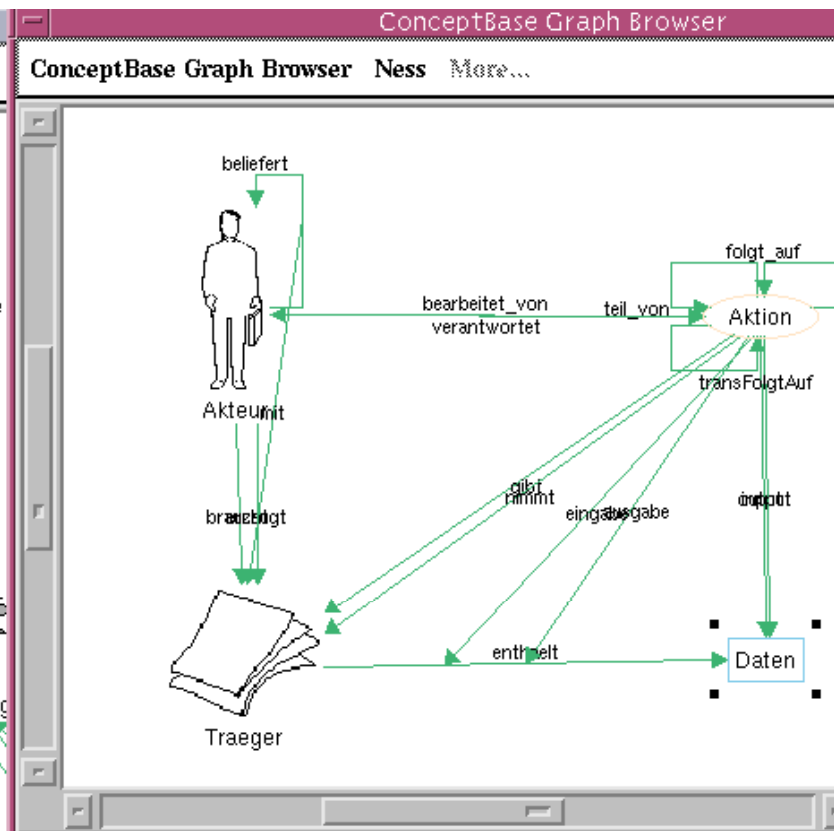
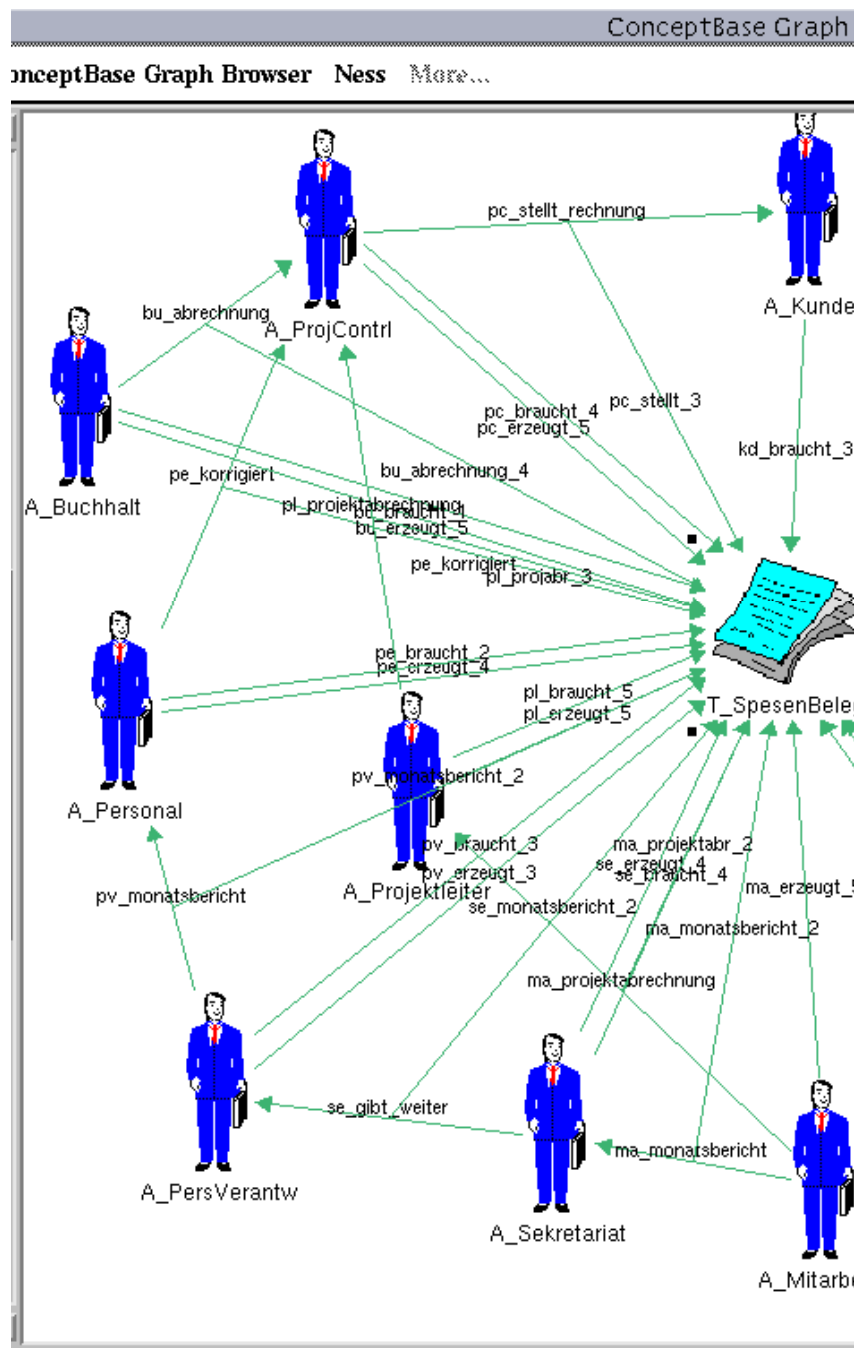
- Experten analysieren die Informationflüsse (ASK)
- Die Anfragen zur Analyse sind "vorfabriziert", können aber jederzeit erweitert werden

Vorteile

- Modellierungssprache (“Metamodell”) ist an die Anwendungssituation anpaßbar (Projekt mit USU: Schwerpunkt auf Datenträger); Anpassung kann ad hoc geschehen
- gemachte Fehler in der Modellierung werden in Form von gespeicherten Anfragen von ConceptBase in “Erfahrung” umgesetzt
- formale Abstützung auf Hornklausellogik: eindeutige Semantik, effiziente Analyse, Konsistenzprüfung

Beispiel einer Anfrage

```
QueryClass QuerZugegriffen isA Daten with
  computed_attribute
    hauptDatum: Daten;
    querProzess: Aktion
  constraint
    c1: $ (hauptDatum aggregiertVon this) and
        (querProzess output this) and
        not (querProzess output hauptDatum) $
  comment
    kommentar1: "Es wird von einer Aktion auf ein Datum this
        zugegriffen, aber nicht auf das uebergeordnete hauptDatum,
        welches this mittelbar oder unmittelbar als Teil
        aggregiert"
end
```



**Ausschnitt aus dem
Geschäftsprozeßmodell
(Daten von USU)**

CBworkbench

Server Edit Browsing Options Help

```

Individual QuerZugegriffen in QueryClass isA Daten with
  computed_attribute,attribute
    hauptDatum : Daten;
    querProzess : Aktion
  constraint,attribute
    c1 : $ (hauptDatum transAggregiertVon this) and
        (querProzess output this) and
        not (querProzess output hauptDatum) $
  comment,attribute
    kommentar1 : "Es wird von einer Aktion auf ein Datum this
        zugegriffen, aber nicht auf das uebergeordnete hauptDatum,
        welches this mittelbar oder unmittelbar als Teil
        aggregiert"
end
        
```

Protocol:

Result: Ok
Successfully connected to port 8888 on tumer !

Load object:
QuerZugegriffen

Result: Answer

```

Individual QuerZugegriffen in QueryClass isA Daten with
  computed_attribute,attribute
    hauptDatum : Daten;
    querProzess : Aktion
  constraint,attribute
    c1 : $ (hauptDatum transAggregiertVon this) and
        (querProzess output this) and
        not (querProzess output hauptDatum) $
  comment,attribute
    kommentar1 : "Es wird von einer Aktion auf ein Datum this
        zugegriffen, aber nicht auf das uebergeordnete hauptDatum,
        welches this mittelbar oder unmittelbar als Teil
        aggregiert"
end
        
```

connected rollback: Now 3 of 3

Display Queries:

- InputOhneNehmen
- InputOhneOutput
- LiefernOhneErzeug
- NieBeliefert
- NimmtNichts
- NimmtOhneBraucht
- OutputOhneAusgabe
- OutputOhneGeben
- P_AlleTraegerGefuelltIn
- P_AlleTraegerNichtGefuelltIn
- P_DatenFalschNutzer
- P_DatenInputNutzer
- P_DatenInputter
- P_DatenNutzer
- P_DatenOutputter
- P_EinAkteurWirdBeliefert
- P_EinAkteurWirdBeliefert2
- P_NachfolgerVon
- P_NichtNachfolgerVon
- P_NichtVorgaengerVon
- P_Rolle
- P_TraegerAusgabe
- P_TraegerEingabe
- P_TraegerGefuelltIn
- P_TraegerNichtGefuelltIn
- P_VorgaengerVon
- QuerZugegriffen**
- TraegerBrauchtOhneErzeugt
- TraegerErzeugtOhneBraucht
- TraegerGegebenListe
- TraegerGenommenListe
- TraegerNichtVerwendet
- TraegerNieErzeugt
- TraegerNieGebraucht
- TraegerOhneDaten
- TraegerZuviel1
- TraegerZuviel2
- UngelieferteTraeger
- WirdMehrfachBeliefert
- exists
- find_instances
- find_references_instances

Ask Telos Editor Cancel Help

Analyse des Geschäftsprozeßmodells

Zusammenfassung

- ConceptBase: eine erweiterbare Metadatenbank für konzeptuelle Modellierungsaufgaben
- saubere Semantik durch Abstützung auf Hornklausellogik
- freie Software (Forschung und nichtkommerzielle Evaluation)
<http://www-i5.informatik.rwth-aachen.de/CBdoc/>
- Validierungsbeispiel: Geschäftsprozeßanalyse
- weltweit ca. 150 Forschungsinstitute und Firmen (British Telecom) nutzen ConceptBase
- neue Funktionen für ConceptBase V5.0
 - Anfrageoptimierer nutzt Information über Kardinalitäten von Klassen
 - Module innerhalb der Objektbank
 - aktive Regeln (Event-Condition-Action)
 - WWW/Java-Schnittstelle

